

## **REMARKS**

[0002] Applicant respectfully requests entry of the following remarks and reconsideration of the subject Application based on this submission under 37 C.F.R. §1.114 and the accompanying fee under 37 C.F.R. §1.17(e). Applicant respectfully requests entry of the amendments to the claims presented above. Additionally, Applicant respectfully request reconsideration and allowance of all of the claims of the Application. Claims 1-15 and 17-38 are presently pending. Claims 1, 9, 12, 14-15, 17-18, 21-23, 26, 34, and 36-38 are amended. Further, claim 16 is canceled and no claims are added.

### **Statement of Substance of Interview**

**[0003]** Examiner Gary Portka graciously talked with me—the undersigned representative for the Applicant—on September 16, 2008. Applicant greatly appreciates the Examiner's willingness to talk. Such willingness is invaluable to both of us in our common goal of an expedited prosecution of this patent application.

**[0004]** During the interview, Examiner Portka and I discussed differences between the Application and the cited references, namely U.S. Patent No. 6,430,576 ("Gates"). Without conceding the propriety of the rejections and in the interest of expediting prosecution, I also proposed several possible clarifying amendments.

**[0005]** Examiner Portka appeared to be receptive to the clarifying amendments and I understood the Examiner to tentatively agree that the clarifying amendments may distinguish the claims over the cited art of record. For example, Examiner Portka indicated that amending the claims to clarify that a policy received at a client from a host includes a number of assertions for the client to comply with in order to access one or more resources via the host, may distinguish the claims over the cited art, namely Gates. However, the Examiner indicated that he would need to review the cited art more carefully and/or do another search, and requested that the proposed amendments be presented in writing.

**[0006]** Applicant amends the claims and presents arguments based on the discussion that occurred during the interview. Accordingly, Applicant submits that

the pending claims are allowable over the cited art of record for at least the reasons discussed during the interview.

**Formal Request for an Additional Interview**

[0007] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an additional interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can discuss this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0008] Please contact me to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for me, I welcome your call as well. My contact information may be found on the last page of this response.

**Claim Amendments**

[0009] Without conceding the propriety of the rejections and in the interest of expediting prosecution, Applicant amends claims 1, 9, 12, 14-15, 17-18, 21-23, 26, 34, and 36-38. The claim amendments are made to expedite prosecution and to more quickly identify allowable subject matter. The amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed features in response to the cited references. The claim amendments are fully supported by the Application and do not include new matter. For example, see page 6, paragraphs [0020] – [0021]; page 19, paragraphs [0059] – [0060]; and pages 25-26, paragraphs [0081] – [0083] of the originally filed Application.

## **SUBSTANTIVE MATTERS**

### **Anticipation Rejections**

**[0010]** Applicant respectfully requests that the Examiner withdraw the anticipation rejections because, for each rejected claim, no single reference discloses each and every element of that rejected claim.<sup>1</sup>

### **Based upon Gates**

**[0011]** The Examiner rejects claims 1, 3, 4, 6-13, 15-18, 20-23, 25, 26, 28, 29 and 31-37 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,430,576 ("Gates"). Applicant respectfully submits that the rejection of claim 16 is moot because claim 16 has been canceled. In addition, Applicant respectfully traverses the rejections of the remaining claims. Based on the reasons given below, Applicant asks the Examiner to withdraw the rejections of these claims.

### **Independent Claim 1**

**[0012]** Applicant submits that the cited portions of Gates do not disclose or show at least the following features of claim 1:

---

<sup>1</sup> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); also see MPEP §2131.

- “receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, and wherein the policy is cached at the client”
- “determining that the client is complying with at least one assertion”
- “generating a policy digest at the client for the cached policy, the policy digest identifying the at least one assertion”

In contrast to claim 1, the cited portions of Gates disclose a “digest” of information indicating changes to an object at a particular client to synchronize with another version of the object at a different client (*See Gates, col. 6, ll. 45-67*) and not a policy digest identifying an assertion that a client is complying with in order to access one or more resources via a host.

**[0013]** Accordingly, claim 1 is allowable because the cited art does not disclose or show each feature of independent claim 1 and Applicant asks the Examiner to withdraw the rejection of this claim.

**Dependent Claims 3-4 and 6-8**

**[0014]** Dependent claims 3-4 and 6-8 ultimately depend upon independent claim 1. As explained previously, the cited art does not disclose or show all of the features of claim 1. Thus, the cited art does not disclose or show all of the features of claims 3-4 and 6-8. Accordingly, claims 3-4 and 6-8 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Independent Claim 9

[0015] Applicant submits that the cited portions of Gates do not disclose or show at least the following features of claim 9:

- “sending a policy from a host to a client, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host”
- “extracting a policy digest from a message received at the host from the client, the policy digest indicating that the client is complying with at least one assertion of the number of assertions”

In contrast to claim 9, the cited portions of Gates disclose a “digest” of information indicating changes to an object at a particular client to synchronize with another version of the object at a different client (*See Gates, col. 6, ll. 45-67*) and not a policy digest indicating that a client is complying with at least one assertion of a number of assertions of a policy in order to access one or more resources via a host.

[0016] Accordingly, claim 9 is allowable because the cited art does not disclose or show each feature of independent claim 9 and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 10-13

**[0017]** Dependent claims 10-13 ultimately depend upon independent claim 9. As explained previously, the cited art does not disclose or show all of the features of claim 9. Thus, the cited art does not disclose or show all of the features of claims 10-13. Accordingly, claims 10-13 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

**Independent Claim 15**

**[0018]** Applicant submits that the cited portions of Gates do not disclose or show at least the following features of claim 15:

- "a message processor to:..."

extract a policy digest from the message, the policy digest indicating that the client is complying with one or more of a number of assertions of a policy in order to access one or more resources via the system and the policy digest including a bit vector identifying the one or more assertions; and

a fault generator to:

return an invalid digest fault to the client when a length of the bit vector is not valid; and

determine whether the one or more assertions are valid when the length of the bit vector is valid"

In contrast to claim 15, the cited portions of Gates disclose a "digest" of information indicating changes to an object at a particular client to synchronize

with another version of the object at a different client (See Gates, col. 6, ll. 45-67) and not a policy digest indicating that a client is complying with one or more of a number of assertions of a policy in order to access one or more resources via a system. In addition, the cited portions of Gates do not disclose or show a policy digest including a bit vector identifying one or more assertions that a client is complying with, as recited in claim 15. Further, the cited portions of Gates do not disclose or show returning an invalid digest fault to a client when a length of the bit vector is not valid and determining whether the one or more assertions are valid when the length of the bit vector is valid, as recited in claim 15.

**[0019]** Accordingly, claim 15 is allowable because the cited art does not disclose or show each feature of independent claim 15 and Applicant asks the Examiner to withdraw the rejection of this claim.

#### Dependent Claims 17-18 and 20

**[0020]** Dependent claims 17-18 and 20 ultimately depend upon independent claim 15. As explained previously, the cited art does not disclose or show all of the features of claim 15. Thus, the cited art does not disclose or show all of the features of claims 17-18 and 20. Accordingly, claims 17-18 and 20 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Independent Claim 21

[0021] Applicant submits that the cited portions of Gates do not disclose or show at least the following features of claim 21:

- "a digest generator to:

generate a policy digest based on one or more policies received at a client from a host, the one or more policies each specifying at least one assertion that the client must comply with in order to access a resource via the host; and

place a bit vector in a header of a message to access a particular resource of the host, the bit vector including one bit for each assertion of a particular policy and including one bit for each assertion of an additional policy referenced by the particular policy"

In contrast to claim 21, the cited portions of Gates disclose a "digest" of information indicating changes to an object at a particular client to synchronize with another version of the object at a different client (See Gates, col. 6, ll. 45-67) and not a policy digest based on one or more policies that each specify at least one assertion that a client must comply with in order to access a resource via a host. In addition, the cited portions of Gates do not disclose or show placing a bit vector in a header of a message to access a particular resource of the host, where the bit vector includes one bit for each assertion of a particular policy and includes one bit for each assertion of an additional policy referenced by the particular policy, as recited in claim 21.

**[0022]** Accordingly, claim 21 is allowable because the cited art does not disclose or show each feature of independent claim 21 and Applicant asks the Examiner to withdraw the rejection of this claim.

**Dependent Claims 22-23 and 25**

**[0023]** Dependent claims 22-23 and 25 ultimately depend upon independent claim 21. As explained previously, the cited art does not disclose or show all of the features of claim 21. Thus, the cited art does not disclose or show all of the features of claims 22-23 and 25. Accordingly, claims 22-23 and 25 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

**Independent Claim 26**

**[0024]** Applicant submits that the cited portions of Gates do not disclose or show at least the following features of claim 26:

- “receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, and wherein the policy is cached at the client”
- “determining that the client is complying with at least one assertion”

- “generating a policy digest at the client for the cached policy, the policy digest identifying the at least one assertion that the client is complying with”

In contrast to claim 26, the cited portions of Gates disclose a “digest” of information indicating changes to an object at a particular client to synchronize with another version of the object at a different client (*See Gates, col. 6, ll. 45-67*) and not a policy digest identifying at least one assertion that a client is complying with in order to access one or more resources via a host.

**[0025]** Accordingly, claim 26 is allowable because the cited art does not disclose or show each feature of independent claim 26 and Applicant asks the Examiner to withdraw the rejection of this claim.

**Dependent Claims 28-29 and 31-33**

**[0026]** Dependent claims 28-29 and 31-33 ultimately depend upon independent claim 26. As explained previously, the cited art does not disclose or show all of the features of claim 26. Thus, the cited art does not disclose or show all of the features of claims 28-29 and 31-33. Accordingly, claims 28-29 and 31-33 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Independent Claim 34

[0027] Applicant submits that the cited portions of Gates do not disclose or show at least the following features of claim 34:

- "extracting at a host a policy digest included in a message from a client, the policy digest indicating that the client is complying with an assertion required to access a resource via the host and the assertion is associated with a policy"

In contrast to claim 34, the cited portions of Gates disclose a "digest" of information indicating changes to an object at a particular client to synchronize with another version of the object at a different client (*See Gates, col. 6, ll. 45-67*) and not a policy digest indicating that a client is complying with an assertion required to access a resources via a host and the assertion is associated with a policy.

[0028] Accordingly, claim 34 is allowable because the cited art does not disclose or show each feature of independent claim 34 and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 35-37

[0029] Dependent claims 35-37 ultimately depend upon independent claim 34. As explained previously, the cited art does not disclose or show all of the features of claim 34. Thus, the cited art does not disclose or show all of the

features of claims 35-37. Accordingly, claims 35-37 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

## **Obviousness Rejections**

### **Lack of *Prima Facie* Case of Obviousness (MPEP § 2142)**

**[0030]** The arguments presented below point to various aspects of the record to demonstrate that all of the criteria set forth for making a *prima facie* case of obviousness have not been met with respect to claims 2, 5, 14, 19, 24, 27, 30, and 38. For example, Applicant respectfully submits that the cited art does not teach or suggest all of the features of claims 2, 5, 14, 19, 24, 27, 30, and 38.

### **Based upon Gates and Atkinson**

**[0031]** The Examiner rejects claims 2, 5, 14, 19, 24, 27, 30 and 38 under 35 U.S.C. § 103(a) as being unpatentable over Gates in view of U.S. Patent No. 6,519,764 ("Atkinson"). Applicant respectfully traverses the rejections of these claims and asks the Examiner to withdraw the rejections of these claims.

### **Dependent Claims 2 and 5**

**[0032]** Dependent claims 2 and 5 depend from claim 1, which Applicant has shown to be allowable over the cited portions of Gates. As explained previously, the cited portions of Gates do not teach or suggest a policy digest identifying an assertion that a client is complying with in order to access one or more resources via a host, as recited in claim 1.

**[0033]** With respect to claims 2 and 5 in view of Atkinson, pages 10 and 11 of the Action state, in part:

"40. Gates does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:

41. As per Claim 2, Atkinson further discloses the method of claim 1, wherein generating the policy digest includes generating a hash of the cached policy [Column 11, Lines 17-23 & Column 28, Lines 39-63]..."

43. As per Claim 5, Atkinson further discloses the method of claim 1, wherein generating the policy digest includes generating a hash of the cached policy if the cached policy is normalized [Column 11, Lines 17-23 & Column 28, Lines 39-63]."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

- This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.
- 5 The following table describes the parameters of the method Hash:

| Argument     | Type    | Description  |
|--------------|---------|--|
| pdwHash      | DWORD * | the place in which to put the returned hash value. |
| return value | HRESULT | S_OK   |

- 5 FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CIItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, II. 39-63).

However, the cited portions of Atkinson do not teach or suggest a policy digest identifying an assertion that a client is complying with in order to access one or more resources via a host, as recited in claim 1.

**[0034]** Since the cited art does not teach or suggest each feature of independent claim 1, the cited art also does not teach or suggest each feature of claims 2 and 5. Accordingly, claims 2 and 5 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Dependent Claim 14

[0035] Dependent claim 14 depends from claim 9, which Applicant has shown to be allowable over the cited portions of Gates. As explained previously, the cited portions of Gates do not teach or suggest a policy digest indicating that a client is complying with at least one assertion of a number of assertions of a policy in order to access one or more resources via a host, as recited in claim 9.

[0036] With respect to claim 14 in view of Atkinson, pages 10 and 11 of the Action state, in part:

"40. Gates does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:....

44. As per Claim 14, Atkinson further discloses the method of claim 9, further comprising reading a row hash of the cached policy [Column 11, Lines 17-23 & Column 28, Lines 39-63]"

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

- This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.
- 5 The following table describes the parameters of the method Hash:

| Argument     | Type    | Description  |
|--------------|---------|--|
| pdwHash      | DWORD * | the place in which to put the returned hash value. |
| return value | HRESULT | S_OK   |

- 5 FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, II. 39-63).

However, the cited portions of Atkinson do not teach or suggest a policy digest indicating that a client is complying with at least one assertion of a number of assertions of a policy in order to access one or more resources via a host, as recited in claim 9.

**[0037]** Since the cited art does not teach or suggest each feature of independent claim 9, the cited art also does not teach or suggest each feature of

claim 14. Accordingly, claim 14 is allowable and Applicant asks the Examiner to withdraw the rejection of this claim.

**Dependent Claim 19**

**[0038]** Dependent claim 19 depends from claim 15, which Applicant has shown to be allowable over the cited portions of Gates. As explained previously, the cited portions of Gates do not teach or suggest a policy digest indicating that a client is complying with one or more of a number of assertions of a policy in order to access one or more resources via a system, as recited in claim 15. In addition, the cited portions of Gates do not teach or suggest a policy digest including a bit vector identifying one or more assertions that a client is complying with, as recited in claim 15, and the cited portions of Gates do not teach or suggest returning an invalid digest fault to a client when a length of the bit vector is not valid and determining whether the one or more assertions are valid when the length of the bit vector is valid, as recited in claim 15.

**[0039]** With respect to claim 19 in view of Atkinson, pages 10 and 11 of the Action state, in part:

"40. Gates does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:...

45. As per Claim 19, Atkinson further discloses the system of claim 15, wherein the policy digest is a row hash of a normalized policy [Column 11, Lines 17-23 & Column 28, Lines 39-63]."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

- This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.
- 5 The following table describes the parameters of the method Hash:

| Argument     | Type    | Description  |
|--------------|---------|--|
| pdwHash      | DWORD * | the place in which to put the returned hash value. |
| return value | HRESULT | S_OK   |

- 5 FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left  
6 hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, ll. 39-63).

However, the cited portions of Atkinson do not teach or suggest a policy digest indicating that a client is complying with one or more of a number of assertions of a policy in order to access one or more resources via a system, a policy digest including a bit vector identifying one or more assertions that the client is complying with, or returning an invalid digest fault to a client when a length of the bit vector is not valid and determining whether the one or more assertions are valid when the length of the bit vector is valid, as recited in claim 15.

**[0040]** Since the cited art does not teach or suggest each feature of independent claim 15, the cited art also does not teach or suggest each feature of claim 19. Accordingly, claim 19 is allowable and Applicant asks the Examiner to withdraw the rejection of this claim.

**Dependent Claim 24**

**[0041]** Dependent claim 24 depends from claim 21, which Applicant has shown to be allowable over the cited portions of Gates. As explained previously, the cited portions of Gates do not teach or suggest a policy digest based on one or more policies that each specify at least one assertion that a client must comply with in order to access a resource via a host, as recited in claim 21. In addition, as explained previously, the cited portions of Gates do not teach or suggest placing a bit vector in a header of a message to access a particular resource of the host, where the bit vector includes one bit for each assertion of a particular policy and includes one bit for each assertion of an additional policy referenced by the particular policy, as recited in claim 21.

**[0042]** With respect to claim 24 in view of Atkinson, pages 10 and 11 of the Action state, in part:

"40. Gates does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:...

46. As per Claim 24, Atkinson further discloses the system of claim 21, wherein generating the policy digest is a row hash of a normalized policy [Column 11, Lines 17-23 & Column 28, Lines 39-63]."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

- This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.
- 5 The following table describes the parameters of the method Hash:

| Argument     | Type    | Description  |
|--------------|---------|--|
| pdwHash      | DWORD * | the place in which to put the returned hash value. |
| return value | HRESULT | S_OK   |

- 5 FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left  
0 hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CltemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, ll. 39-63).

However, the cited portions of Atkinson do not teach or suggest a policy digest based on one or more policies that each specify at least one assertion that a client must comply with in order to access a resource via a host or placing a bit vector in a header of a message to access a particular resource of the host, where the bit vector includes one bit for each assertion of a particular policy and includes one bit for each assertion of an additional policy referenced by the particular policy, as recited in claim 21.

**[0043]** Since the cited art does not teach or suggest each feature of independent claim 21, the cited art also does not teach or suggest each feature of claim 24. Accordingly, claim 24 is allowable and Applicant asks the Examiner to withdraw the rejection of this claim.

**Dependent Claims 27 and 30**

**[0044]** Dependent claims 27 and 30 depend from claim 26, which Applicant has shown to be allowable over the cited portions of Gates. As explained previously, the cited portions of Gates do not teach or suggest a policy digest identifying an assertion that a client is complying with in order to access one or more resources via a host, as recited in claim 26.

**[0045]** With respect to claims 27 and 30 in view of Atkinson, pages 10, 11, and 12 of the Action state, in part:

"40. Gates does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:....

47. As per Claim 27, Atkinson further discloses the computer program product of claim 26 wherein the computer process further comprises generating a hash of the cached policy [Column 11, Lines 17-23 & Column 28, Lines 39-63]

48. As per Claim 30, Atkinson further discloses the method of claim 26, wherein the computer process further comprises generating a row hash of

the cached policy if the cached policy is normalized [Column 11, Lines 17-23 & Column 28, Lines 39-63]."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

- This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.
- 5 The following table describes the parameters of the method Hash:

| Argument     | Type    | Description  |
|--------------|---------|--|
| pdwHash      | DWORD * | the place in which to put the returned hash value. |
| return value | HRESULT | S_OK   |

- 5 FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, II. 39-63).

However, the cited portions of Atkinson do not teach or suggest a policy digest identifying an assertion that a client is complying with in order to access one or more resources via a host, as recited in claim 26.

**[0046]** Since the cited art does not teach or suggest each feature of independent claim 26, the cited art also does not teach or suggest each feature of claims 27 and 30. Accordingly, claims 27 and 30 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Dependent Claim 38

[0047] Dependent claim 38 depends from claim 34, which Applicant has shown to be allowable over the cited portions of Gates. As explained previously, the cited portions of Gates do not teach or suggest a policy digest indicating that a client is complying with an assertion required to access a resources via a host and the assertion is associated with a policy, as recited in claim 34.

[0048] With respect to claim 38 in view of Atkinson, pages 10 and 12 of the Action state in part:

"40. Gates does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:...

49. As per Claim 38, Atkinson further discloses the computer program product of claim 34 wherein the computer process further comprises reading a row hash of the cached policy if the cached policy is normalized [Column 11, Lines 17-23 & Column 28, Lines 39-63]."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.

The following table describes the parameters of the method Hash:

| Argument     | Type    | Description  |
|--------------|---------|--|
| pdwHash      | DWORD * | the place in which to put the returned hash value. |
| return value | HRESULT | S_OK   |

FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, II. 39-63).

However, the cited portions of Atkinson do not teach or suggest a policy digest indicating that a client is complying with an assertion required to access a resources via a host and the assertion is associated with a policy, as recited in claim 34.

**[0049]** Since the cited art does not teach or suggest each feature of independent claim 34, the cited art also does not teach or suggest each feature

of claim 38. Accordingly, claim 38 is allowable and Applicant asks the Examiner to withdraw the rejection of this claim.

### **Conclusion**

**[0050]** All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call or email me at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC  
Representatives for Applicant

/Trevor Lind/ Dated: October 6, 2008

Trevor Lind ([trevor@leehayes.com](mailto:trevor@leehayes.com); 512-505-8165)

Registration No. 54785

Christopher Lattin ([christopher@leehayes.com](mailto:christopher@leehayes.com); 509-324-9256, ext. 263)

Registration No. 56064

Customer No. **22801**

Facsimile: (509) 323-8979

[www.leehayes.com](http://www.leehayes.com)